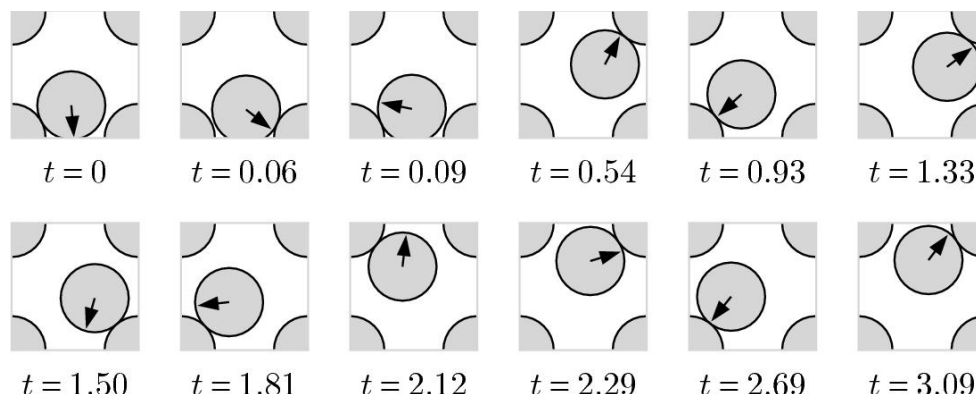


Class Session 2: The Markov matrix and the exponential convergence

Ergodicity

Modern research in mathematics has gone far in actually proving ergodicity, the equivalence between Newton's deterministic mechanics and Boltzmann's statistical mechanics, for the special case of hard spheres. The first milestone result of Sinai (1970), a 50-page proof, shows rigorously that the two-disk problem is ergodic. Several decades later it has become possible to prove that general hard-disk and hard-sphere systems are indeed ergodic, under very mild assumptions (Simanyi 2003, 2004). Here we recover numerically Sinai's result.



References

- Sinai Y. G., Dynamical systems with elastic reflections, Russian Mathematical Surveys 25, 137-189 (1970) (<http://iopscience.iop.org/0036-0279/25/2/R05>)
- Simanyi N., Proof of the Boltzmann-Sinai ergodic hypothesis for typical hard disk systems, Inventiones Mathematicae 154, 123-178 (2003) (<http://www.springerlink.com/content/yf9uw55m3vm6utqx/>)

Sinai's Problem

Consider Sinai's system of two large disks in a square box of size L with periodic boundary conditions. The radius of the disks is $L/4 < \sigma$

1. Show that in the reference frame of a stationary disk (remaining at position $(0,0)$), the center of the moving disk reflects off the stationary one as in geometric optics, with the incoming angle equal to the outgoing angle.
2. Implement the time evolution of this system, with stops at regular time intervals.
3. Compute the two dimensional histogram of positions, $\pi(x, y)$, and determine from it the histogram of projected densities.

Algorithm

```

In [9]: import numpy, math
        from pylab import *
        %matplotlib inline

        def pair_time(x,vec,v):
            del_x = numpy.subtract(x,vec)
            dummy = numpy.dot(del_x,v)
            Upsilon = dummy**2 - (numpy.dot(del_x,del_x) - 4*sigma**2)
            if Upsilon > 0 and dummy < 0 :
                t_pair = - (dummy + sqrt(Upsilon))
            else:
                t_pair = 1e20
            return t_pair
        sigma = 0.26
        statistics = 400000
        # initial conditions
        x = numpy.array([0.5,0.5])
        v = numpy.array([1,0.])
        corner = [[0,0],[1,0],[0,1],[1,1]]
        t = 0.
        last_corner = 0
        x_bin = [x[0]]
        y_bin = [x[1]]
        for iter in range(statistics):
            next_time = [pair_time(x,vec,v) for vec in corner]
            t_1 = min(next_time)
            corner_1 = corner[next_time.index(t_1)]
            if math.floor(t) != math.floor(t+t_1):
                dummy = x + v*(math.floor(t+t_1)-t)
                x_bin.append(dummy[0])
                y_bin.append(dummy[1])
            x += v*(t_1)
            t += t_1
            e_perp = numpy.subtract(corner_1,x)
            e_perp = e_perp/sqrt(dot(e_perp,e_perp))
            v -= 2*e_perp*numpy.dot(v,e_perp)
        hexbin(x_bin,y_bin, gridsize=40)
        axis([0.,1.,0.,1.])
        title("Histogram of positions in Sinai's problem")
        cb = colorbar()
        cb.set_label('counts')
        show()

```

